

(19)日本国特許庁 (JP)

(12) 特許公報 (B2)

(11)特許番号

特許第3086779号

(P3086779)

(45)発行日 平成12年9月11日(2000.9.11)

(24)登録日 平成12年7月7日(2000.7.7)

(51)Int.Cl.⁷

G 0 6 F 15/177

識別記号

6 8 2

F I

G 0 6 F 15/177

6 8 2 L

請求項の数5(全12頁)

(21)出願番号 特願平7-151736

(22)出願日 平成7年6月19日(1995.6.19)

(65)公開番号 特開平9-6731

(43)公開日 平成9年1月10日(1997.1.10)

審査請求日 平成9年3月5日(1997.3.5)

(73)特許権者 000003078

株式会社東芝

神奈川県川崎市幸区坂川町72番地

(72)発明者 増渕 美生

東京都青梅市末広町2丁目9番地 株式会社東芝青梅工場内

(74)代理人 100058479

弁理士 鈴江 武彦

審査官 石井 茂和

最終頁に続く

(54)【発明の名称】 メモリ状態復元装置

(57)【特許請求の範囲】

【請求項1】 データ処理を実行する少なくとも1つのプロセッサと、
前記プロセッサにおけるデータ処理に必要なデータ等を記憶するための主メモリと、
前記プロセッサに対応して設けられたデータ一貫性機能を有するコピーバック型のキャッシュと、
前記主メモリにおけるアドレスと同アドレスで指定されるロケーションに記憶されるデータとを組にして格納するためのバッファメモリと、
前記キャッシュから発行されるトランザクションを監視して、前記プロセッサから対応する前記キャッシュに対してライトアクセス要求した場合に、当該キャッシュからキャッシュラインの無効化を指示する無効化トランザクションが発行されたことを開始条件として、前記ライ

トアクセス要求が起きた前記主メモリ上のアドレス、及びこのアドレスで指定されるロケーションに格納されたデータを、前記主メモリから読み出し前記バッファメモリに保存するメモリアクセス制御手段とを具備したことを特徴とするメモリ状態復元装置。

【請求項2】 データ処理を実行する複数のプロセッサと、

前記プロセッサにおけるデータ処理に必要なデータ等を記憶するための主メモリと、

前記プロセッサに対応して設けられたデータ一貫性機能を有するコピーバック型のキャッシュと、

前記主メモリにおけるアドレスと同アドレスで指定されるロケーションに記憶されるデータとを組にして格納するためのバッファメモリと、

前記キャッシュから発行されるトランザクションを監視

して、前記プロセッサから対応する前記キャッシュに対してライトアクセス要求してキャッシュミスをした場合に、無効化を伴うリードトランザクションが発行されたことを開始条件とし、他のキャッシュから更新データを保持する信号が output されない時、前記ライトアクセス要求が起きた前記主メモリ上のアドレス、及びこのアドレスで指定されるロケーションに格納されたデータを、前記主メモリから読み出し前記バッファメモリに保存し、また読み出しだデータを前記キャッシュへ出力するメモリアクセス制御手段とを具備したことを特徴とするメモリ状態復元装置。

【請求項3】前記プロセッサがリードアクセス要求した結果、キャッシュミスした場合に発行されるリードトランザクションに応答して、他のキャッシュからキャッシュミスしたキャッシュラインのクリーンデータ又は更新データを保持している信号が発行されたならば、前記他のキャッシュから前記主メモリに書き出されるキャッシュラインデータを取り込み、当該キャッシュラインをクリーンシェアード状態に設定する手段と、前記プロセッサが前記クリーンシェアード状態のキャッシュラインにライトアクセス要求を行った場合、データ書き込みと共に前記無効化トランザクションを発行する手段とを有することを特徴とする請求項1または2記載のメモリ状態復元装置。

【請求項4】前記バッファメモリは、前記主メモリの一部を利用することを特徴とする請求項1または2記載のメモリ状態復元装置。

【請求項5】チェックポイント時には前記キャッシュに保持されている更新データを前記主メモリに書き出す処理だけを実行することを特徴とする請求項1または2記載のメモリ状態復元装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、メモリの内容を以前の状態に戻すことを可能とする機能を有するコンピュータシステムのメモリ状態復元装置に関する。

【0002】

【従来の技術】通常の計算機では、プログラムを実行した場合に、一旦処理が進んでしまうと、それ以前の状態に戻って処理を再開するということが、一般にできない。ところが、次のような応用においては、メモリの内容を以前の状態に戻してその時点から処理を継続する機能（メモリ状態回復機能）があることが望まれる。

【0003】(1) ソフトウェアデバッグ

プログラムの実行中に何らかのエラーが発生した場合に、以前の状態に戻ることにより、エラーの原因を解析することができる。

【0004】(2) フォールトトレラントシステム

システムの動作中に何らかの故障により処理が続行できなくなった場合、以前の状態に戻って、そこから処理を

再開することにより、システムを停止させることなく動作を継続することができる。

【0005】このようなフォールトトレラント技術は、例えばPhilip A Bernstein, "Sequoia : A Fault-Tolerant Tightly Coupled Multiprocessor for Transaction Processing," IEEE Computer, Vol. 21, No. 2, 1988に開示されている。

【0006】(3) バックトラッキング

論理型のプログラミング言語では、実行状態のバックトラッキングが基本的な操作である。メモリの内容が以前の状態に戻る機能を用いることにより、バックトラッキングを実現することができる。

【0007】以上のようなメモリ状態回復機能を実現する方法として考えられている技術の一つに、「バックワード手法」がある。図5は、バックワード手法を採用するシステムの一つの構成例を示した図である。図5に示すシステムは、プロセッサ30、メモリ制御部31、主メモリ32、及びビフォアイメージバッファ33から構成される。

【0008】ビフォアイメージバッファ33は、メモリ制御部31の制御のもとに、主メモリ32の以前の状態を保持するためのメモリであり、一つのエントリ（「ビフォアイメージエレメント」と呼ぶ）は主メモリアドレスとデータからなる。

【0009】図5のように構成されたシステムの動作例を以下に説明する。いま、プロセッサ30が主メモリ32のあるロケーションAに、データDnewをライトする場合を考える。

【0010】メモリ制御部31は、プロセッサ30から「Write」処理の要求を受けると、主メモリ32を更新する前に、同じロケーションAに格納されていたデータDoldを読み出し、ロケーションのアドレス値Aとともに、ビフォアイメージバッファ33に格納する。その後、メモリ制御部31は、データDnewを主メモリ32のロケーションAに書き込む。

【0011】メモリ制御部31は、プロセッサ30から「Write」処理の要求を受ける毎にこの動作を繰り返し、主メモリ32のアドレスとデータとをビフォアイメージバッファ33の別なエントリに順次格納していく。

【0012】主メモリ32を以前の状態に戻したい場合には、メモリ制御部31は、ビフォアイメージバッファ33に格納されたエントリを最新のものから順次読み出し（アドレスA、データDold）、アドレスAで示されるメモリロケーションに、データDoldを書き込んで行く。これにより、主メモリ32を以前の状態に戻すことができる。

【0013】なお、一般に、プログラムをある状態から再開させるためには、以前の主メモリ32の内容だけでなく、プロセッサ30の内部状態も必要である。プロセ

ッサ30の内部状態を保存するための一つの方法としては、適当な時間間隔で内部状態を主メモリ32に保存しておくチェックポイント方式がある。ここで、チェックポイント方式では、保存するタイミングを「チェックポイント」、主メモリ32の内容及びプロセッサ30の内部状態を保存することを「チェックポイントをとる」という。

【0014】チェックポイントをとる際には、同時にビフォアイメージバッファ33をクリアする。従って、ビフォアイメージバッファ33には、最新のチェックポイント時点から現在までに更新された主メモリ32のロケーション（アドレス）の元の値が保存されていることになる。

【0015】これにより、プログラムは任意の時点から最新のチェックポイント時点に戻すことが可能となる。このような技術は、例えばRok Sosic, 「History Cache: Hardware Support for Reverse Execution,」 Computer Architecture News, Vol. 22, No. 5, 1994に開示されている。

【0016】次に、マルチプロセッサシステムに対して、前述のようなメモリ状態回復機能を適用する例について説明する。図6は、n個のプロセッサ30-1～30-nがバス34を介して接続された形態のマルチプロセッサシステムである。メモリ制御部31は、バス34を介して各プロセッサ30-1～30-nからの処理要求を受け付ける。

【0017】図6に示すマルチプロセッサシステムにおいても、メモリ制御部31、主メモリ32、及びビフォアイメージバッファ33の動作は、図5に示す構成の例と同様に制御することができる。

【0018】すなわち、メモリ制御部31は、プロセッサ30-1～30-nから「Write」処理の要求を受ける毎に、主メモリ32を更新する前に主メモリ32から該当するデータDoldを読み出して、アドレスと共にビフォアイメージバッファ33に順次格納していく。

【0019】主メモリ32を以前の状態に戻したい場合には、メモリ制御部31は、同様にビフォアイメージバッファ33に格納されたエントリを最新のものから順次読み出し、そのアドレスAで示されるメモリロケーションに、データDoldを書き込んで行く。

【0020】また、適当な時間間隔でチェックポイントをとり、全てのプロセッサ30-1～30-nの内部状態を保存することにより、任意の時点からチェックポイントの時点に戻り、処理を再開させることができる。

【0021】ところで、近年のプロセッサは、メモリアクセスの高速化のためにキャッシュメモリを持つことが一般的となっている。キャッシュメモリには、ライトスルー型のキャッシュとコピーバック型のキャッシュの二種類がある。

【0022】ライトスルー型のキャッシュの場合は、ブ

ロセッサがライト処理を実行したとき、キャッシュに記憶されたデータの値を更新すると共に、同時に主記憶に記憶されたデータもキャッシュに保持された値に更新する。従って、プロセッサが持つキャッシュの内容と主メモリの内容とが一致しているので、メモリ状態回復機能は、前に述べた方法と同様の技術で実現できる。チェックポイント処理についても、全く同様の方法で実現できる。

【0023】一方、コピーバック型のキャッシュの場合は、プロセッサがライト処理を実行したとき、更新されるのはキャッシュ中の値だけであり、主メモリには更新された内容が直ぐに反映されない。その後、キャッシュエントリのリプレースなどによって更新されたデータが主メモリに書き出されることにより、初めて主メモリの内容が更新される。なお、キャッシュの内容を主メモリに書き出す場合（「Write-Line」処理）、一般的には複数のワードから構成されるキャッシュライン単位で行なわれる。

【0024】図7は、n個のプロセッサ30-1～30-nに、それぞれキャッシュ40-1～40-nが設けられた形態のマルチプロセッサシステムである。図7に示すマルチプロセッサシステムは、キャッシュ40-1～40-nがコピーバックキャッシュであった場合、メモリ状態回復機能を実現するために次のように動作する。

【0025】チェックポイント時には、図8(a)に示すように、プロセッサの内部状態だけでなく、キャッシュ中に保持されている主メモリ32に反映されていない全ての更新データ（「A」「B」「C」）が主メモリ32に書き戻されて、このチェックポイントでのシステムの状態が保存され、その後、ビフォアイメージバッファ33がクリアされる。

【0026】また、チェックポイント後にキャッシュからメモリ制御部31に「Write-Line」処理の要求が発行された場合、つまりキャッシュ上で更新されたデータ（「a」を含むキャッシュライン）を主メモリ32に反映させる場合には、図8(b)に示すように、メモリ制御部31は、チェックポイントにおいて保存したデータの内容を確保しておくために、ビフォアイメージバッファ33にデータを移す。

【0027】すなわち、メモリ制御部31は、キャッシュから「Write-Line」処理の要求を受けると、主メモリ32の書き戻しの対象となるロケーションAlineからラインデータDold-line（データ「A」を含む）を読み出し、アドレス値Alineと共にビフォアイメージバッファ33に格納する（ビフォアイメージバッファ33に格納される一つのエントリは、主メモリ32におけるラインアドレスとラインデータから構成される）。その後、メモリ制御部31は、キャッシュ上の更新データ（データ「a」を含むキャッシュライン）を主メモリ

32に書き戻す。

【0028】主メモリ32の内容を以前の状態(直前のチェックポイント時の状態)に戻す場合には、メモリ制御部31は、ビフォアイメージバッファ33に格納されたエントリを最新のものから順次読み出して(アドレスAline、ラインデータDold-line)、そのラインアドレスAlineで示されるメモリロケーションに、ラインデータDold-lineを書き込んで行く。これにより、主メモリ32を以前のチェックポイント時の状態に戻すことができる(ここでは、主メモリ32中のデータについてのみ考へている)。

【0029】ところで、チェックポイント時には、キャッシュ上で更新され、主メモリ32に反映されていない全ての更新データを、一括して主メモリ32に書き戻す必要がある。従って、集中して多数の「Write-Line」処理が、メモリ制御部31に対して発行されることになる。また、主メモリ32にデータが書き戻されるために、主メモリ32に存在する書き戻しの対象となる古いデータも、一括してビフォアイメージバッファ33に格納しなければならない。

【0030】以上のようなメモリ状態回復機能では、一つの「Write-Line」処理について、主メモリ32に対するリード及びライトの2回のアクセス、及びビフォアイメージバッファ33に対するライトアクセスが必要となるため、「Write-Line」処理が集中して発生するチェックポイントでは、大量のメモリアクセスが発生する。

【0031】大量のメモリアクセスが発生するチェックポイントでは、その間システムが停止したようになり、他の通常の処理ができなくなってしまうため、チェックポイント処理に多くの時間を要してしまうと、システム全体の処理効率の低下を招いてしまう。

【0032】このような傾向は、プロセッサの数が増大するにつれ、またキャッシュの容量が増大してチェックポイント時に書き出すべきキャッシュライン数が増大するにつれて顕著になる。従って、大規模かつ高性能なシステムを構成するためには、大きな問題となる。

【0033】

【発明が解決しようとする課題】このように従来のコンピュータシステムでは、コピーバック型キャッシュを用いたマルチプロセッサシステムにおいて、チェックポイントをとる際に、キャッシュ上の更新された状態で記憶されている全てのデータが一括して主メモリ32に書き戻されるため、それに伴って主メモリ32中の該当する各データをビフォアイメージバッファ33に保存する必要となってしまう。このため、チェックポイント処理に要する時間が増大し、チェックポイントの処理の間に他の処理が実行できないことにより、システム性能が劣化してしまうという問題があった。

【0034】本発明は前記のような事情を考慮してなさ

れたもので、チェックポイント処理に要する時間を短縮してシステム性能を向上させることができマルチプロセッサシステム等のコンピュータシステムを提供することを目的とする。

【0035】

【課題を解決するための手段】本発明は、データ処理を実行する少なくとも1つのプロセッサと、前記プロセッサにおけるデータ処理に必要なデータ等を記憶するための主メモリと、前記プロセッサに対応して設けられたデータ一貫性機能を有するコピーバック型のキャッシュと、前記主メモリにおけるアドレスと同アドレスで指定されるロケーションに記憶されるデータとを組にして格納するためのバッファメモリと、前記キャッシュから発行されるトランザクションを監視して、前記プロセッサから対応する前記キャッシュに対してライトアクセス要求した場合に、当該キャッシュからキャッシュラインの無効化を指示する無効化トランザクションが発行されたことを開始条件として、前記ライトアクセス要求が起きた前記主メモリ上のアドレス、及びこのアドレスで指定されるロケーションに格納されたデータを、前記主メモリから読み出し前記バッファメモリに保存するメモリアクセス制御手段とを具備したことを特徴とする。

【0036】また、本発明は、データ処理を実行する複数のプロセッサと、前記プロセッサにおけるデータ処理に必要なデータ等を記憶するための主メモリと、前記プロセッサに対応して設けられたデータ一貫性機能を有するコピーバック型のキャッシュと、前記主メモリにおけるアドレスと同アドレスで指定されるロケーションに記憶されるデータとを組にして格納するためのバッファメモリと、前記キャッシュから発行されるトランザクションを監視して、前記プロセッサから対応する前記キャッシュに対してライトアクセス要求してキャッシュミスをした場合に、無効化を伴うリードトランザクションが発行されたことを開始条件とし、他のキャッシュから更新データを保持する信号が输出されない時、前記ライトアクセス要求が起きた前記主メモリ上のアドレス、及びこのアドレスで指定されるロケーションに格納されたデータを、前記主メモリから読み出し前記バッファメモリに保存し、また読み出しだデータを前記キャッシュへ出力するメモリアクセス制御手段とを具備したことを特徴とする。

【0037】また、前記プロセッサがリードアクセス要求した結果、キャッシュミスした場合に発行されるリードライントランザクションに応答して、他のキャッシュからキャッシュミスしたキャッシュラインのクリーンデータ又は更新データを保持している信号が発行されたならば、前記他のキャッシュから前記主メモリに書き出されるキャッシュラインデータを取り込み、当該キャッシュラインをクリーンシェアード状態に設定する手段と、前記プロセッサにより前記クリーンシェアード状態のキ

キャッシングにライトアクセス要求が行われた場合、データ書き込みと共に前記無効化トランザクションを発行する手段とを有することを特徴とする。また、前記バッファメモリは、前記主メモリの一部を利用するすることを特徴とする。

【0038】

【作用】あるプロセッサがあるアドレスのデータを更新する際に、無効化要求を含むトランザクションが発行される。このとき、主メモリからそのアドレスで示されるデータを読みだし、これをアドレスとともにバッファメモリ（ビフォーアイメージバッファ）に格納する。この処理は、実際にそのデータを、キャッシングから主メモリに書き出すよりも以前に先行して実行される。これにより、チェックポイント時にはキャッシングに保持されていた更新データを主メモリに書き出すだけで良くなり、オーバヘッドが小さくなる。

【0039】

【実施例】以下、図面を参照して本発明の一実施例を説明する。図1は本実施例のコンピュータシステム（マルチプロセッサシステム）の概略構成を示すものである。図1に示すように、本実施例におけるマルチプロセッサシステムは、n個のプロセッサ10-1～10-n、各プロセッサに対応するn個のキャッシング12-1～12-nが設けられている。キャッシング12-1～12-nは、メモリ制御部14とバス16を介して接続されている。メモリ制御部14には、主メモリ17、及びビフォーアイメージバッファ18が接続されている。

【0040】メモリ制御部14は、バスインタフェース制御部141、状態保存制御部142、主メモリアクセス制御部143、及びバッファアクセス制御部144を有している。

【0041】バスインタフェース制御部141は、バス16と直接接続されており、バスの制御を行なう。バスインタフェース制御部141は、常時、バス16を監視しており、バス16に発行されたトランザクションに応じて動作する。状態保存制御部142は、主メモリ17の状態を保存するために必要な各種機能を制御する。主メモリアクセス制御部143は、主メモリ17と接続され、主メモリ17に対するアクセスを制御する。バッファアクセス制御部144は、ビフォーアイメージバッファ18と接続され、ビフォーアイメージバッファに対するアクセスを制御する。

【0042】キャッシング12-1～12-nは、コピーバック型のキャッシングであり、以下に説明する方式に従って、データ一貫性保持のためのプロトコルを実現する。キャッシング12-1～12-nは、キャッシングラインデータを保持するデータメモリと、データメモリに格納された各キャッシングラインデータを管理する情報を保持するためのタグメモリから構成される。

【0043】図2に、タグメモリのエントリの一例を示

す。タグメモリには、対応するキャッシングラインデータのキャッシングラインアドレス（ADDR）を示すアドレスタグと、キャッシングラインデータの状態を示す3ビットのデータが格納される。

【0044】キャッシングラインデータの状態は、バリッド（V: Valid）、モディファイド（M: Modified）、シェアード（S: Shared）の3ビットの値の組合せにより管理される。なお、バリッド（V）は、対応するキャッシングラインデータが有効（“1”）であることを示す。モディファイド（M）は、キャッシング上で更新された状態にあることを示す。シェアード（S）は、他のプロセッサのキャッシング上にも同じキャッシングラインデータが保持されていることを示す。

【0045】従って、3ビットの値の組合せにより、図3に示すように、キャッシングラインデータは4つの状態を取る。コピーバックキャッシングに関連した、キャッシングからバス16に発行されるトランザクションは、次のようなものを含む。

【0046】(1) 「Read-Line」…キャッシングラインデータのリード。これは、プロセッサからの、あるアドレスのキャッシングラインに対するリードアクセスに対して、該当する有効なキャッシングラインデータが、キャッシングに存在せずキャッシングミスした場合に発行される。

【0047】なお、キャッシングミスは、アクセスの対象とするキャッシングラインデータを示すアドレスが、タグメモリに格納されていない、及びアドレスは格納されているがバリッドビット（V）が無効（“0”）を示す場合である。これに対し、キャッシングヒットは、対象とするキャッシングラインデータのアドレスがタグメモリに格納され、バリッドビット（V）が有効（“1”）を示す場合である。

【0048】(2) 「Read-Line-with-Invalid」…キャッシングラインデータのリード、及び他のキャッシングに格納されたデータ無効化。これは、プロセッサからのライトアクセスに対して、キャッシングミスした場合に発行される。

【0049】(3) 「Write-Line」…キャッシングラインデータのライト。これは、プロセッサからの指示により明示的にキャッシングラインデータを主メモリ17に書き出す場合、キャッシングラインデータを置換する必要が生じて更新データが主メモリ17に書き出される場合、及び他のプロセッサからの要求に応じて更新データを主メモリ17に書き出す場合に発行される。

【0050】(4) 「Invalid」…他のキャッシングのデータ無効化。これは、プロセッサからのライトアクセスに対して、キャッシングに当該ラインが存在したが、クリーンシェアードであった場合に発行される。このトランザクション自身はアドレスの転送のみであり、データ転送を伴わない。

【0051】なお一般に、コピーバックキャッシュに関連したトランザクションは、前述した(1)～(4)以外のトランザクションもサポートされることが多いが、本発明の趣旨には直接関係ないので、ここでは説明を省略する。

【0052】バス16は、シェアード応答信号線161、モディファイド応答信号線162、バスコマンド信号線163、アドレス/データ信号線164を含む。一般には、この他にアービトリエーションを行なうための信号線などを含むが、本発明の趣旨には直接関係ないので、ここでは説明を省略する。

【0053】シェアード応答信号線161(shared)は、他のプロセッサ/キャッシュにより発行されたトランザクションに対して、トランザクションが対象としているキャッシュラインデータを保持している(すなわち共有)していることを通知するために用いられる。

【0054】モディファイド応答信号線162(modified)は、他のプロセッサ/キャッシュにより発行されたトランザクションに対して、トランザクションが対象としているキャッシュラインデータを更新された状態で保持していることを通知するために用いられる。

【0055】バスコマンド信号線163(command)は、前述したバストランザクション(1)～(4)の種類を示す他、トランザクションに関する各種情報を送るために用いられる。

【0056】アドレス/データ信号線164(address/data)は、バストランザクションが対象とするキャッシュラインデータのアドレス及びデータを送るため信号線である。図1においては1本の信号線として示しているが、共通の信号線を時分割で使う方式や、アドレス/データ各々に独立した信号線を設ける方式があり、何れであっても構わない。

【0057】次に、本実施例におけるマルチプロセッサシステムの動作、すなわちバストランザクションを用いて、キャッシュ間のデータ一貫性保持プロトコルがどのように実現されるかを説明する。

【0058】ここでは、トランザクションを発行してアクセスを起動する側のプロセッサ及びキャッシュ(マスタプロセッサ)、あるプロセッサ/キャッシュが発行したトランザクションに対して動作する他のプロセッサ及びキャッシュ(スレーブプロセッサ)、及びメモリ制御部14のそれぞれの動作について説明する。

【0059】まず、トランザクションを発行してアクセスを起動するマスタプロセッサの動作と、発行されたトランザクションに対応するキャッシュの状態遷移について説明する。なお、マスタプロセッサがプロセッサ10-1(キャッシュ12-1)であるとして説明する。

【0060】(A1) リードアクセス(キャッシュヒット)

プロセッサ10-1がリードアクセス要求した結果、キ

ャッシュ12-1においてキャッシュヒットした場合、キャッシュ12-1から該当するデータが読み出される。プロセッサ10-1(キャッシュ12-1)は、バス16にトランザクションを発行しない。この時、キャッシュラインの状態は変わらない。

【0061】(A2) リードアクセス(キャッシュミス)

プロセッサ10-1がリードアクセス要求した結果、キャッシュミスした場合、プロセッサ10-1(キャッシュ12-1)は、バス16に「Read-Line」トランザクションを発行する。

【0062】これに対して、モディファイド応答信号線162を介してモディファイド応答信号がアサートされたならば、他のキャッシュのうち一つがそのラインの更新データを保持していることになる(スレーブプロセッサ側の動作の詳細については後述する)。キャッシュ12-1は、モディファイド応答信号をアサートしたキャッシュから主メモリ17に書き出される(アドレス/データ信号線164に出力される)キャッシュラインデータを取り込んで、データメモリに格納する。キャッシュラインの状態は「クリーン・シェアード」とする。

【0063】一方、モディファイド応答信号がアサートされず、シェアード応答信号線161を介してシェアード応答信号がアサートされたならば、他のキャッシュがそのラインをクリーンな状態で保持していることになる。キャッシュ12-1は、キャッシュラインの状態を「クリーン・シェアード」とすると共に、主メモリ17より読み出されたデータを取り込んで、データメモリに格納する。

【0064】また、モディファイド応答信号とシェアード応答信号の何れもアサートされない場合には、どのキャッシュにも対象とするキャッシュラインデータが保持されていないことになる。キャッシュ12-1は、キャッシュラインの状態を「クリーン・エクスクルーシブ」とする。この場合も、キャッシュ12-1は、主メモリより読み出されたデータを取り込み、データメモリに格納する。

【0065】何れの場合も、キャッシュ12-1は、バス16から取り込んだキャッシュラインデータのうち、必要とされるデータをプロセッサ10-1に返す。

(A3) ライトアクセス(キャッシュヒット/モディファイド)

プロセッサ10-1がライトアクセス要求した結果、キャッシュ12-1においてキャッシュヒットし、対応するキャッシュラインデータがモディファイドの状態の場合、該当するキャッシュラインにデータが書き込まれる。プロセッサ10-1(キャッシュ12-1)は、バス16にトランザクションを発行しない。この時、キャッシュラインの状態は変わらない。

【0066】(A4) ライトアクセス(キャッシュヒッ

ト／クリーンエクスクルーシブ)

プロセッサ10-1がライトアクセス要求した結果、キャッシュ12-1においてキャッシュヒットし、対応するキャッシュラインデータがクリーンエクスクルーシブの状態の場合、該当するキャッシュラインにデータが書き込まれる。プロセッサ10-1(キャッシュ12-1)は、バス16にトランザクションを発行しない。キャッシュ12-1は、該当するキャッシュラインの状態を「モディファイド」に変更する。

【0067】(A5) ライトアクセス(キャッシュヒット／クリーンシェアード)

プロセッサ10-1がライトアクセス要求した結果、キャッシュ12-1においてキャッシュヒットし、対応するキャッシュラインデータがクリーンシェアードの状態の場合、該当するキャッシュラインにデータが書き込まれる。プロセッサ10-1(キャッシュ12-1)は、バス16に「`Invalid`」トランザクションを発行する。キャッシュ12-2は、該当するキャッシュラインの状態を「モディファイド」にする。

【0068】(A6) ライトアクセス(キャッシュミス)

プロセッサ10-1がライトアクセス要求した結果、キャッシュミスした場合、バスに「`Read-Line-with-Invalid`」を発行する。

【0069】これに対して、モディファイド応答信号がアサートされたならば、他のキャッシュのうち一つがそのラインの更新データを保持することになる。キャッシュ12-1は、モディファイド応答信号をアサートしたキャッシュから主メモリ17に書き出されるキャッシュラインデータを取り込んで、データメモリに格納する。

【0070】一方、モディファイド応答信号がアサートされなければ、キャッシュ12-1は、主メモリ17より読み出されたデータ取り込み、データメモリに格納する。何れの場合も、キャッシュ12-1は、該当するキャッシュラインの状態を「モディファイド」とし、データを書き込む。

【0071】次に、あるプロセッサ／キャッシュが発行したトランザクションに対する他のプロセッサとそれに応するキャッシュ(スレーブプロセッサ)の動作、及び状態遷移について説明する。なお、マスタプロセッサがプロセッサ10-1(キャッシュ12-1)、スレーブプロセッサがプロセッサ10-n(キャッシュ12-n)であるとし、マスタプロセッサから発行されるトランザクション毎に説明する。

【0072】(B1)「`Read-Line`」トランザクション

キャッシュ12-nは、トランザクションが対象とするキャッシュラインデータを「モディファイド」の状態で保持している場合、モディファイド応答信号162をアサ

ートして、更新データを保持していることを通知する。この後、キャッシュ12-nは、「`Write-Line`」トランザクションを発行して、更新データを主メモリ17に書き出す。キャッシュ12-nは、該当するキャッシュラインの状態を「クリーンシェアード」とする。

【0073】また、キャッシュ12-nは、トランザクションが対象とするキャッシュラインデータを「クリーンエクスクルーシブ」または「クリーンシェアード」の状態で保持している場合、シェアード応答信号161をアサートして、クリーンデータを保持していることを通知する。何れの場合も、キャッシュラインの状態は「クリーンシェアード」とする。

【0074】なお、キャッシュ12-nは、対応する有効なキャッシュラインデータを保持していない場合には、何もしない。

(B2)「`Read-Line-with-Invalid`」トランザクション

キャッシュ12-nは、トランザクションが対象とするキャッシュラインデータを「モディファイド」の状態で保持している場合、モディファイド応答信号162をアサートして、更新データを保持していることを通知する。この後、キャッシュ12-nは、「`Write-Line`」トランザクションを発行して更新データを主メモリ17に書き出す。キャッシュラインの状態は「インバリッド」とする。

【0075】また、キャッシュ12-nは、トランザクションが対象とするキャッシュラインデータを「クリーンエクスクルーシブ」または「クリーンシェアード」の状態で保持している場合、キャッシュラインの状態を「インバリッド」とする。

【0076】なお、キャッシュ12-nは、対応する有効なキャッシュラインデータを保持していない場合には、何もしない。

(B3)「`Invalid`」トランザクション
キャッシュ12-nは、トランザクションが対象とするキャッシュラインデータを「クリーンエクスクルーシブ」または「クリーンシェアード」の状態で保持している場合、このキャッシュラインの状態を「インバリッド」にする。

【0077】また、キャッシュ12-nは、対応する有効なキャッシュラインデータを保持していない場合には、何もしない。なお、この場合、対応するラインを「モディファイド」の状態で保持していることはあり得ない。

【0078】(B4)「`Write-Line`」トランザクション

キャッシュ12-nは、何もしない。次に、各バストランザクションに対するメモリ制御部14の動作について説明する。メモリ制御部14の主メモリ17の状態を保

存するための動作制御は、状態保存制御部142によつて行なわれる。以下、バス16に発行されるトランザクション毎に説明する。

【0079】(C1) 「Write-Line」トランザクション

バスインタフェース制御部141がバス16に発行された「Write-Line」トランザクションをスヌープすると、状態保存制御部142は、主メモリアクセス制御部143を起動する。主メモリアクセス制御部143は、キャッシュから書き出されるキャッシュラインデータを、バスインタフェース制御部141を介して取り込み、主メモリ17の該当するアドレスに書き込む。

【0080】(C2) 「 Invalidate」トランザクション

バスインタフェース制御部141がバス16に発行された「Invalidate」トランザクションをスヌープすると、状態保存制御部142は、主メモリアクセス制御部143を起動する。主メモリアクセス制御部143は、バスインタフェース制御部141を介して得た、無効対象となつてゐるキャッシュラインのアドレスで示されるメモリロケーションから更新前のキャッシュラインデータを読み出す。また、状態保存制御部142は、バッファアクセス制御部144を起動する。状態保存制御部142は、主メモリアクセス制御部143によつて読み出されたキャッシュラインデータをバッファアクセス制御部144に転送して、アドレス値と共にビフォアイメージバッファ18に書き込ませる。

【0081】(C3) 「Read-Line-with-Invalidate」トランザクション

「Read-Line-with-Invalidate」トランザクションの発行に伴つて、モディファイド応答信号がアサートされた場合は、アサートしたキャッシュが既に更新データを保持していることを示しており、そのプロセッサが最新のチェックポイント以降にライトアクセスを実行していることがわかる。すなわち、前述した(A5)のように、ライトアクセスが実行される際に、バス16に「Invalidate」トランザクションが発行され、前述した(C2)の処理が既に実行されていることになる。このため、その時点では、以前のデータがビフォアイメージバッファ18に格納されていることになり、改めてこのアドレスのデータをビフォアイメージバッファに格納する必要はない。従つて、状態保存制御部142は、何もする必要がない。

【0082】一方、「Read-Line-with-Invalidate」トランザクションの発行に伴つて、モディファイド応答信号がアサートされない場合は、状態保存制御部142は、バスインタフェース制御部141から通知を受け、この通知に応じて主メモリアクセス制御部143を起動する。主メモリアクセス制御部143は、バスインタフェース制御部141を介して

得た、無効対象としてアドレスで示される主メモリ17のメモリロケーションからキャッシュラインデータを読み出す。バスインタフェース制御部141は、主メモリアクセス制御部143によつて読み出されたラインデータをバス16に出力する。また、状態保存制御部142は、バッファアクセス制御部144を起動して、主メモリアクセス制御部143によつて読み出されたキャッシュラインデータをバッファアクセス制御部144に転送し、アドレス値と共にビフォアイメージバッファ18に書き込ませる。

【0083】(C4) 「Read-Line」トランザクション

バスインタフェース制御部141が「Read-Line」トランザクションの発行に伴つて、モディファイド応答信号がアサートされたことをスヌープすると、状態保存制御部142は、主メモリアクセス制御部143を起動する。主メモリアクセス制御部143は、キャッシュから書き出されるラインデータをバスインタフェース制御部141を介して取り込み、主メモリ17の該当するアドレスに書き込む。

【0084】一方、モディファイド応答信号162がアサートされない場合は、状態保存制御部142は、主メモリアクセス制御部143を起動する。主メモリアクセス制御部143は、バスインタフェース制御部141を介して得た、読み出し対象としてアドレスで示される主メモリ17のメモリロケーションからキャッシュラインデータを読み出す。バスインタフェース制御部141は、主メモリアクセス制御部143によつて読み出されたラインデータをバス16に出力する。また、ラインデータの出力と共に、状態保存制御部142は、バスインタフェース制御部141を介して、シェアード応答信号をシェアード応答信号線161にアサートする。これによつて、マスタプロセッサのキャッシュの、現在対象としているキャッシュラインの状態を「クリーンエクスクルーシブ」ではなく、「クリーンシェアード」の状態にすることができる。

【0085】ここで、「クリーンエクスクルーシブ」状態を避け、「クリーンシェアード」の状態にする理由は、次の通りである。マスタプロセッサについての説明のよう(前述した(A4))、「クリーンエクスクルーシブ」の状態にあるキャッシュラインに対してライトする場合には、バス16にトランザクションが発行されることなくキャッシュ中で値が書き換えられてしまう。このため、この時点ではビフォアイメージバッファ18に、主メモリ17への書き込みを行なう以前のメモリ状態(対象とするキャッシュラインについて)を保存することができない。

【0086】これに対応するためには、キャッシュから主メモリ17にラインデータが書き出されたとき、すなわち「Write-Line」トランザクションがバス

16に発行されたときに、メモリ状態を保存する方法が考えられる。しかし、これでは結局、従来と同様になってしまい、チェックポイント処理時間が長くなるという問題がある。

【0087】「クリーンエクスクルーシブ」状態がなければ、ライトアクセスする場合には、必ずバス16に「*Invalidate*」または「*Read-Line-with-Invalidate*」の何れかのトランザクションが発行される。メモリ制御部14は、このトランザクションを、主メモリ17からビフォアイメージバッファ18にデータを移す処理の開始条件として、上述のようにして書き込み以前のメモリ状態を保存することが可能となる。

【0088】なお、プロセッサ10-1～10-n、及びキャッシュ12-1～12-nに「クリーンエクスクルーシブ」状態を避ける機能が設けられている場合には、メモリ制御部14は、シェアード応答信号161をアサートする必要はない。

【0089】次に、前述のようにして動作するマルチプロセッサシステムにおけるチェックポイント処理について説明する。チェックポイント時には、プロセッサ10-1～10-nの内部状態を主メモリ17に書き出すとともに、各キャッシュ12-1～12-nの「モディファイド」状態にある全てのキャッシュラインのデータを主メモリ17に書き戻す。キャッシュ12-1～12-nは、主メモリ17に書き戻したキャッシュラインの状態を「クリーンシェアード」または「インバリッド」にする。

【0090】主メモリ17へのラインデータの書き戻しは、「*Write-Line*」トランザクションを用いて行なわれるため、このときにビフォアイメージバッファ18へのデータ保存が発生することはない。

【0091】すなわち、チェックポイント処理を開始する以前に、キャッシュ12-1～12-n上で更新されたデータに対応する主メモリ17中のデータが、前述した(C2) (C3) ように既に、ビフォアイメージバッファ18に保存されているので、主メモリ17への書き込み処理によりチェックポイント処理が完了する。

【0092】図4に、本発明を用いるマルチプロセッサシステムの他の構成例を示す。図4に示す構成では、主メモリ26の記憶容量の一部を、ビフォアイメージバッファ28として使用するものである。

【0093】バッファアクセス制御部244は、主メモリアクセス制御部243を介して、主メモリ26中のビフォアイメージバッファ28をアクセスする。その他の制御は、図1に示す構成に対するものと同様である。

【0094】図4に示す構成をとると、ビフォアイメージバッファ用に主メモリとは別の特別なメモリを必要としないため、システム構成を簡単することができるという効果がある。

【0095】なお、本発明はコピーバック型キャッシングを対象としているが、コピーバックモードと非コピーバックモードのデータが混在するコンピュータシステムに対しても当然適用可能である。すなわち、コピーバックモードのデータに対しては本発明を適用し、非コピーバックモードのデータに対しては従来例の方法を適用すれば良い。

【0096】例えば、前記実施例においては、複数のプロセッサ10-1～10-nを持つマルチプロセッサシステムを説明したが、キャッシングが同様の機能を持っていれば、單一プロセッサから成るコンピュータシステムに対しても適用することが可能である。

【0097】また、キャッシングが單一でなく、階層構造をとる場合にも同様に適用可能である。このようにして、メモリ制御部14は、あるキャッシングからバス16に発行された、他のキャッシングに保持されているデータを無効化するためのトランザクションがスヌープされた場合に、主メモリ17に保持された無効化の対象となるキャッシングラインデータをビフォアイメージバッファ18に保存する。

【0098】すなわち、マスタプロセッサでのライトアクセス要求時にキャッシングヒットであり、対象とするキャッシングラインの状態がクリーンシェアードの場合に発行された「*Invalidate*」、あるいはライトアクセス要求時にキャッシングミスした場合に発行された「*Read-Line-with-Invalidate*」に応じて、主メモリ17中のキャッシングラインデータがビフォアイメージバッファ18に保存されるので、対象とするキャッシングラインデータを、実際に主メモリ17に書き戻す処理の以前に実行されることになる。

【0099】従って、チェックポイント時に、主メモリ17に書き戻す必要のある更新データが大量にあったとしても、既に書き戻しの対象となるデータがビフォアイメージバッファ18に保存されているので、そのための処理が不要となる。このために、チェックポイント処理に要する時間が短縮され、他の通常の処理に対する影響が低減されるので、システム全体の処理効率の向上が図れる。

【0100】

【発明の効果】以上詳述したように本発明によれば、チェックポイントによって保持された、以前のメモリ状態を保存する処理が、チェックポイント時に集中するのではなく、通常の処理の途中に分散して行なわれるのでは、チェックポイント時のオーバヘッドが軽減され、処理効率を向上させることが可能となるものである。

【図面の簡単な説明】

【図1】本発明の一実施例に係わるコンピュータシステム(マルチプロセッサシステム)の概略構成を示すブロック図。

【図2】本実施例におけるキャッシングタグエントリの一

例を説明するための図。

【図3】本実施例におけるキャッシュラインの状態を説明するための図。

【図4】本発明におけるマルチプロセッサシステムの他の構成例を示すブロック図。

【図5】従来のバックワード手法を採用するシステムの構成例を示すブロック図。

【図6】従来のメモリ状態回復機能を説明するためのマルチプロセッサの構成を示すブロック図。

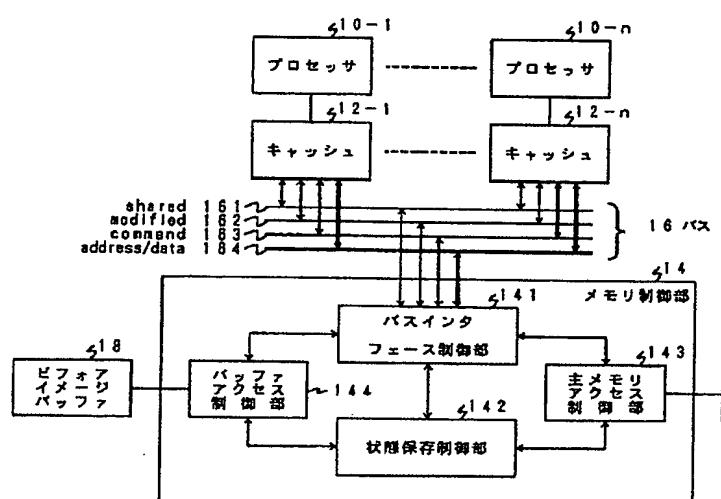
【図7】従来のメモリ状態回復機能を説明するためのキャッシュが設けられたマルチプロセッサの構成を示すブロック図。

【図8】従来のメモリ状態回復機能を説明するための図。

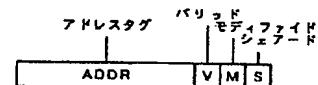
【符号の説明】

10-1～10-n…プロセッサ、12-1～12-n…キャッシュ、14…メモリ制御部、141…バスインターフェース制御部、142…状態保存制御部、143…主メモリアクセス制御部、144…バッファアクセス制御部、16…バス、161…シェアード応答信号線、162…モディファイド応答信号線、163…バスコマンド信号線、164…アドレス/データ信号線、17…主メモリ、18…ビフォアイメージバッファ。

【図1】



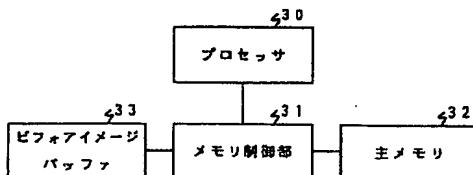
【図2】



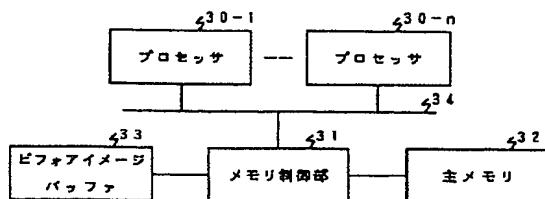
【図3】

| パリティ | モディファイド | シェアード | 状態 |
|------|---------|-------|---------------|
| V | M | S | |
| 0 | — | — | インパリティ |
| 1 | 0 | 0 | クリーン・エクスクルーシブ |
| 1 | 0 | 1 | クリーン・シェアード |
| 1 | 1 | 0 | モディファイド |

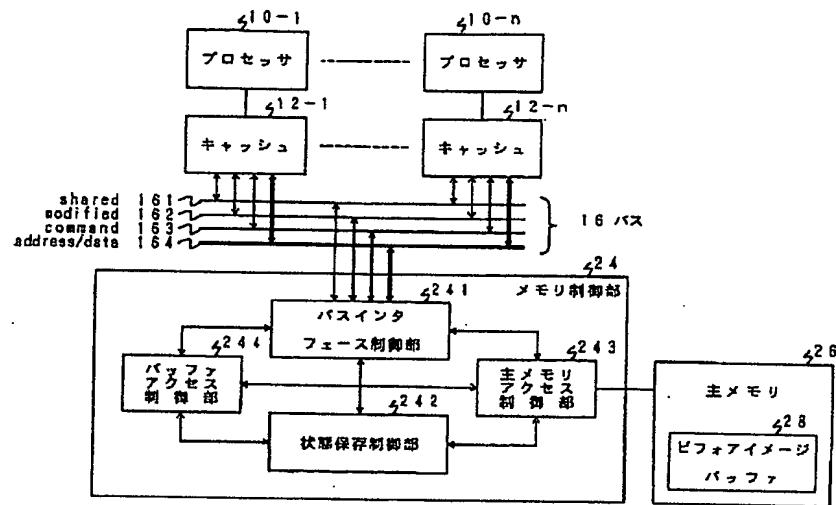
【図5】



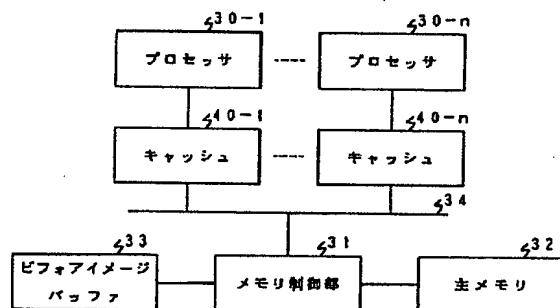
【図6】



【図4】

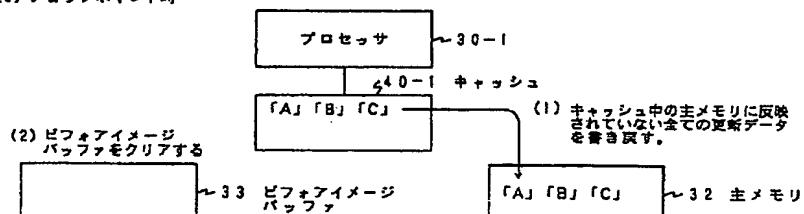


【図7】

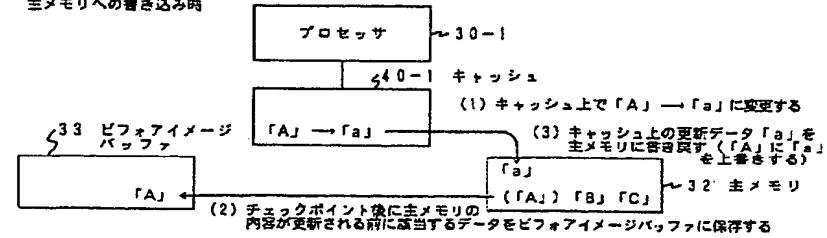


【図8】

(a) チェックポイント時



(b) チェックポイント後の主メモリへの書き込み時



フロントページの続き

(56)参考文献 米国特許5418916 (U.S., A)
 IEEE TRANSACTIONS
 ON PARALLEL AND D
 ISTRIBUTED SYSTEMS
 VOL. 1 NO. 2 1990 p231
 —240 Kun-Lung Wu, W.
 Kent Fuchs & Janak
 H. Patel [Error Rec
 overy in Shared Me
 mory Multiprocessor
 s Using Private C
 aches]
 1990 IEEE 20th Intern
 ational Symposium
 On Fault-Tolerant
 Computing p82—88
 Rana Ejaz Ahmed, R
 obert C. Frazier &
 Peter N. Marinos [Ca
 che-Aided Rollback
 Error Recovery (CA
 RER) Algorithms for
 Shared-Memory Mul
 tiprocessor System
 s]
 IEEE TRANSACTIONS
 ON PARALLEL AND D
 ISTRIBUTED SYSTEMS
 VOL. 5 NO. 10 1994 p1033
 —1043 Bob Janssens &
 W. Kent Fuchs [The
 Performance of Cac
 he-Based Error Rec
 overy in Multiproc
 essors]
 1987 FAULT-TOLERANT
 COMPUTING p170—175
 Douglas B. Hunt &
 Peter N. Marinos [A
 General Purpose Ca
 che-Aided Rollback
 Error Recovery (CA
 RER) Technique]

(58)調査した分野(Int.Cl., DB名)
 G06F 15/177 682
 INSPEC (DIALOG)
 JICSTファイル (JOIS)
 WPI (DIALOG)